ORACLE®

ORACLE®

MySQL Developer Essentials with PHP, Java and .NET

Ralf Gebhardt
Principal Sales Engineer

Agenda

- MySQL Overview
- Development Basics
- Java
- PHP
- .NET
- Resources and Q&A

About MySQL

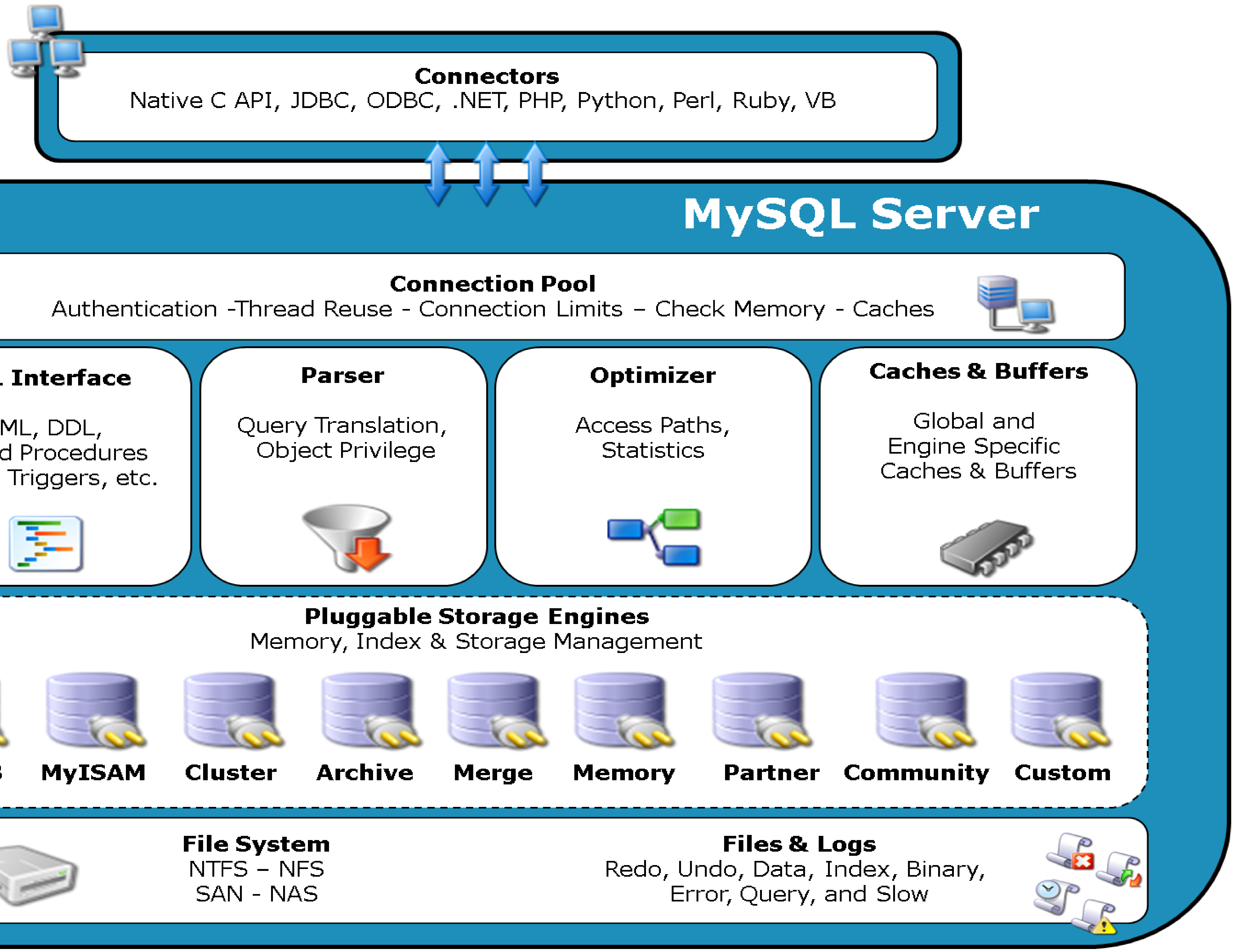
- 15 years of development
- Acquired by Sun in February 2008
- Acquired by Oracle in January 2010
- #1 Most popular Open Source Database
  - Low Cost
  - Easy to Use
  - Performance, Reliability and Scalability

Oracle's Plans for MySQL

- Fill-In Oracle's database product suite
- MySQL Global Business Unit
- Invest in MySQL
  - "Make MySQL a Better MySQL"
  - Develop, promote and support MySQL
- MySQL Community Edition
  - Source and binary releases
  - GPL license

ORACLE

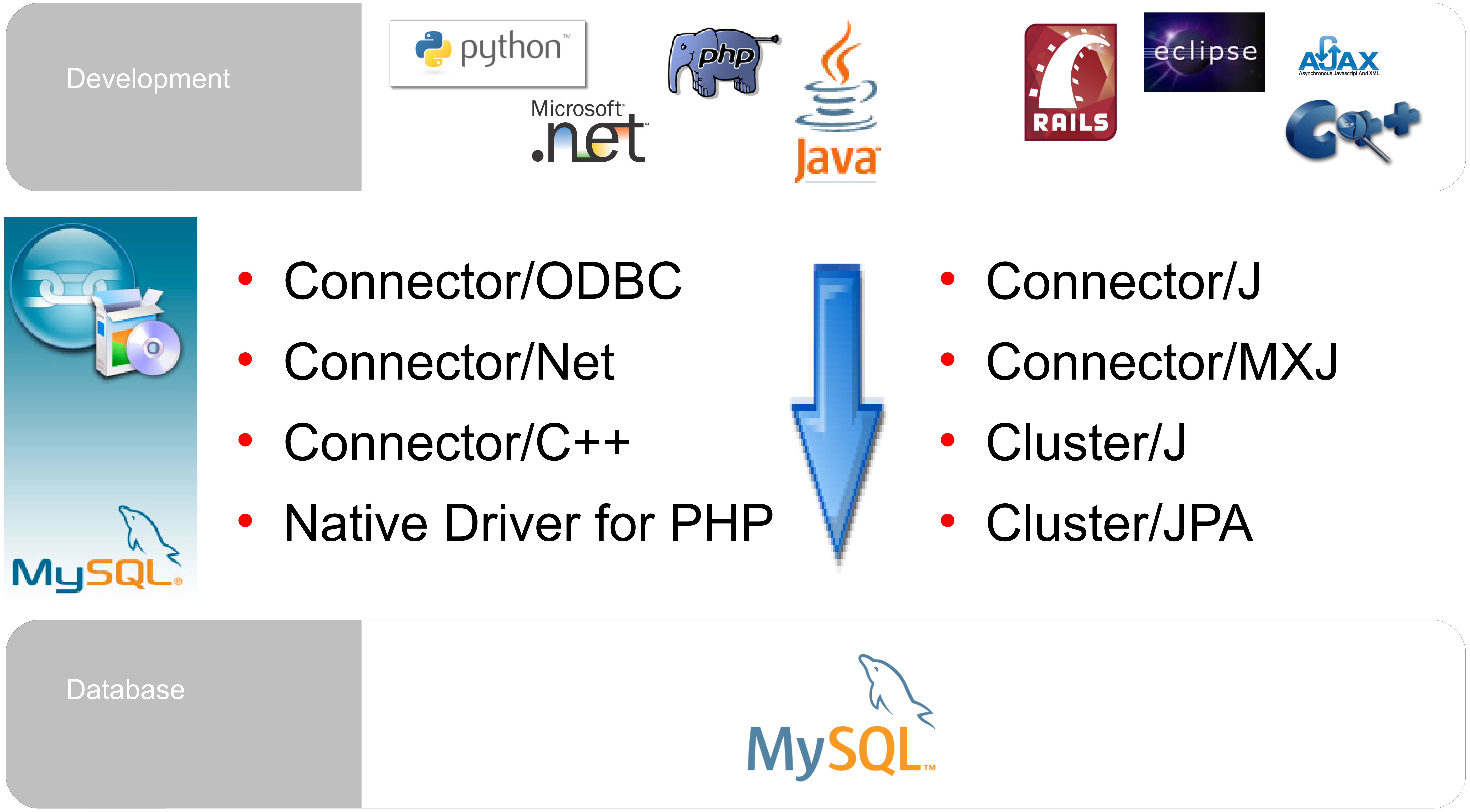
MySQL Architecture



Connectors
Native C API, JDBC, ODBC, .NET, PHP, Python, Perl, Ruby, VB
MySQL Server
Connection Pool
Authentication -Thread Reuse - Connection Limits – Check Memory - Caches
SQL Interface
DML, DDL, Stored Procedures Views, Triggers, etc.
Parser
Query Translation, Object Privilege
Optimizer
Access Paths, Statistics
Caches & Buffers
Global and Engine Specific Caches & Buffers
Pluggable Storage Engines
Memory, Index & Storage Management
InnoDB
MyISAM
Cluster
Archive
Merge
Memory
Partner
Community
Custom
File System
NTFS – NFS
SAN - NAS
Files & Logs
Redo, Undo, Data, Index, Binary, Error, Query, and Slow

You need to know the differences

MySQL Connectors



- Connector/ODBC
- Connector/Net
- Connector/C++
- Native Driver for PHP

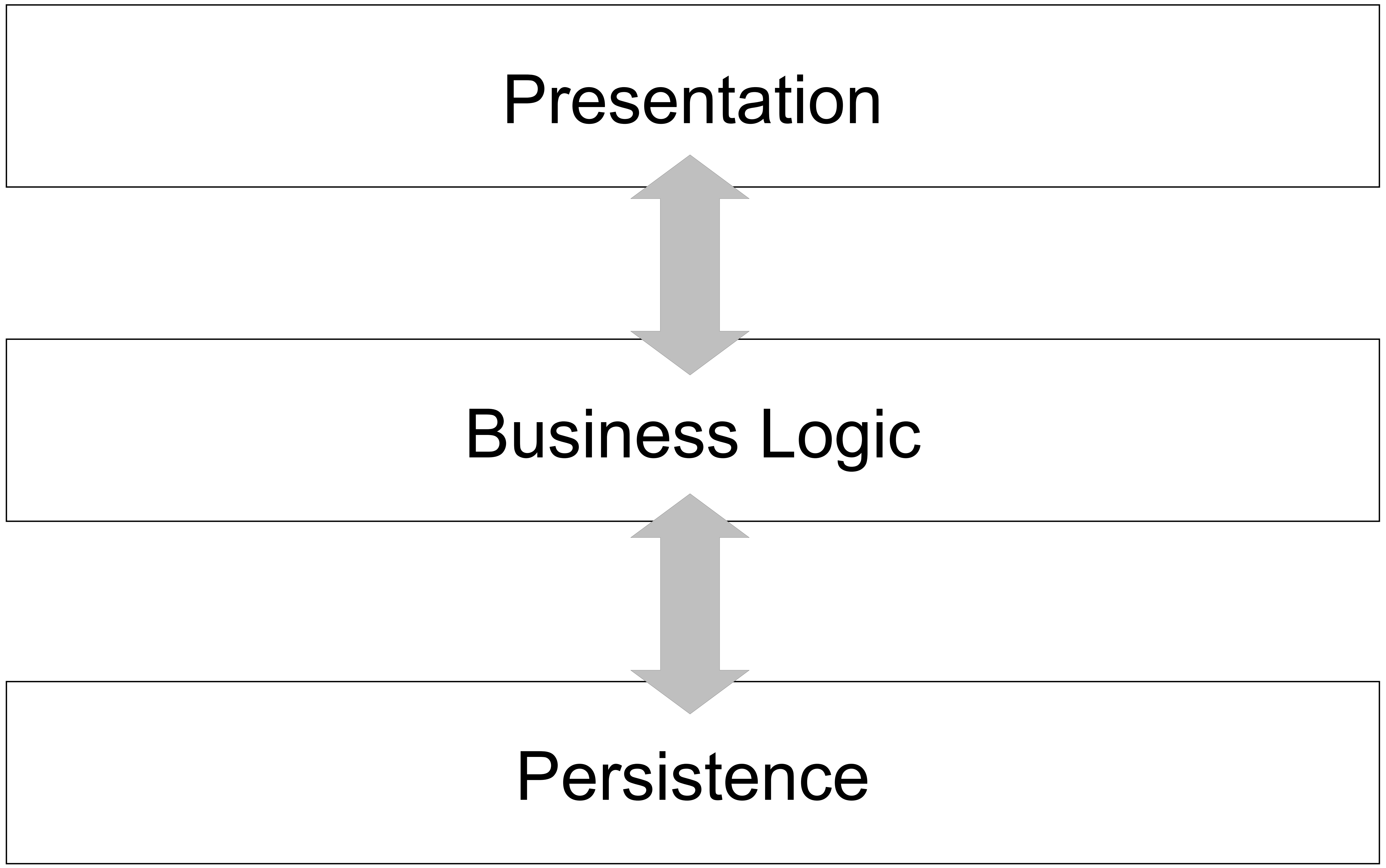
- Connector/J
- Connector/MXJ
- Cluster/J
- Cluster/JPA

Development Basics

Web Application Primer – Focus on the business logic

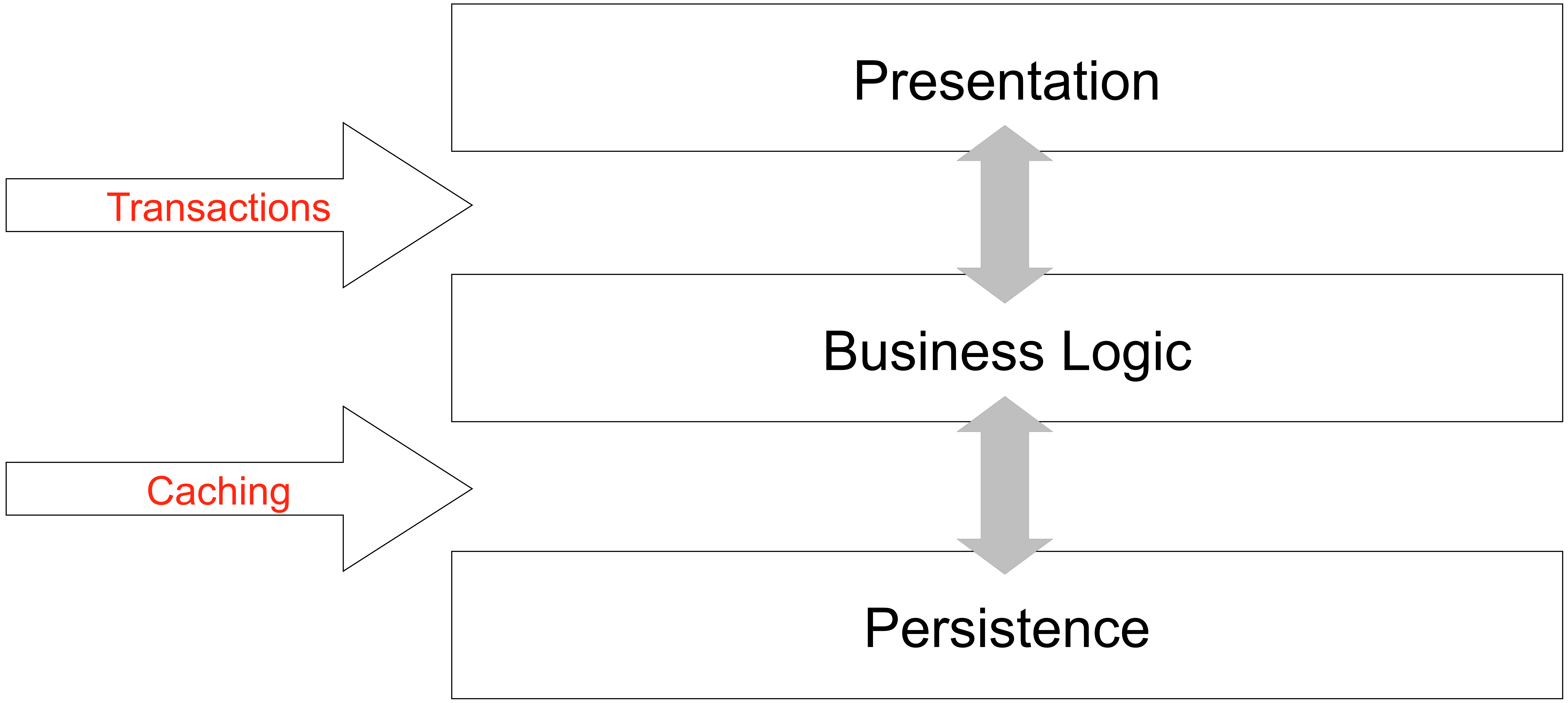
- Clean separation of presentation, business logic, persistence
  - See also MVC (Model View Controller) Architecture
- Find and use the appropriate framework
  - Don't reinvent the wheel!
- Look for the framework to provide anything non-business related
  - Beware if you start refactoring (or worse, cut and pasting) application plumbing

Architecting for Success - Start Clean

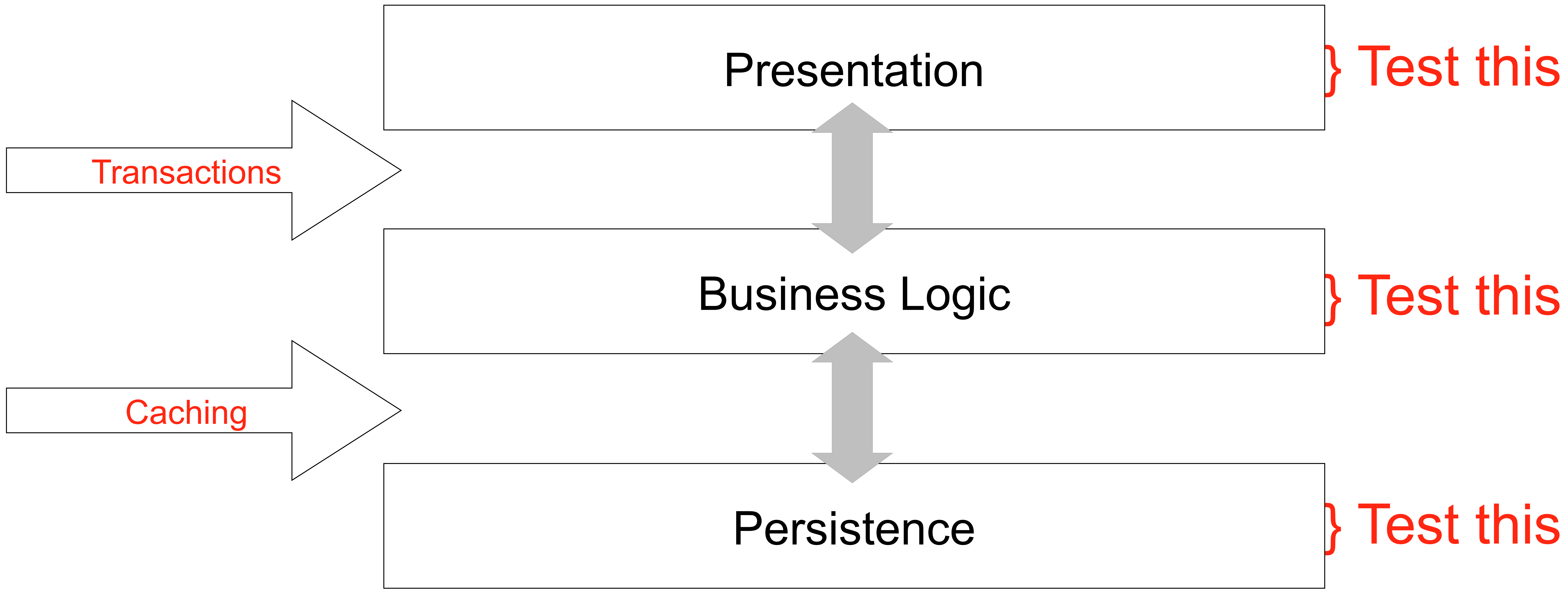


ORACLE

Architecting for Success - Start Clean



Architecting for Success - Start Clean



Interacting with the database

- Keep DB-related code together (ideally in the framework)
  - Helps performance tuning
    - Leverage new features, simplify configuration
  - Helps portability
    - You don't want to track down vendor-specific keywords in 1000s of files
- Start vendor neutral (vanilla SQL), add vendor-specific code as you weigh it's worth

Java

Connecting Java to MySQL

- The JDBC driver is called MySQL Connector/J
- Type IV (all-java)
- Available from the following sources
  - Software
    - http://dev.mysql.com/downloads/connector/j/5.1.html
    - Maven, Ivy, Ant & Spring are all projects you should get familiar with
  - Many Linux and BSD distributions and Solaris all have Oracle Java software (including Connector/J) available in their repositories
- Documentation
  - http://dev.mysql.com/doc/refman/5.1/en/connector-j.html

More Java Resources

- Use MySQL with Java
  http://dev.mysql.com/usingmysql/java/
- Read Connector/J User Manual
  http://dev.mysql.com/doc/refman/5.5/en/connector-j.html
- Visit MySQL “JDBC and Java” Forum
  http://forums.mysql.com/list.php?39

Leveraging the Frameworks – Don't use raw JDBC!

- An example with the Spring framework and Java annotations:

```
@Override

public User createUser(String login, String firstName, String lastName, String
credentials) {

jdbcTemplate.update("INSERT INTO user (login, first_name, last_name, passwd_hash) VALUES
(?, ?, ?, ?)", login, firstName, lastName, credentials);

long id = jdbcTemplate.queryForLong("SELECT LAST_INSERT_ID()");

PersistentUser user = new PersistentUser();

user.setId(id);

user.setLogin(login);

user.setFirstName(firstName);

user.setLastName(lastName);

user.setCredentials(credentials);

return user;

}
```

...versus raw JDBC

```
public User createUser(Connection c, String login,
String firstName, String lastName, String
credentials) {

PreparedStatement p = null;


try {

p = c.prepareStatement("INSERT INTO user (login,
first_name, last_name, passwd_hash) VALUES
(?, ?, ?, ?)");

p.setString(1, login);

p.setString(2, firstName);

......

ResultSet rs = p.getGeneratedKeys();

rs.next();

long id = rs.getLong(1);

}
```

```
PersistentUser user = new PersistentUser();

user.setId(id);

......

return user;

} catch (SQLException sqlEx) {

// handle it, it's not a concern outside of
persistence

} finally {

try {

if (p != null) { p.close(); };

} catch (SQLException sqlEx) {

// can't do anything here, log?

}

}

return null;
```

Extensions

- HA and Clustering
  - Add multiple hosts to the connect string for replication/cluster awareness
    - jdbc:replication://master,slave1,slave2,...,slaveN/
  - Auto failover (readonly for Master-Slave)
  - Auto-load balance among slaves for non-DML
    - Requires your code knows when to create RO vs. RW statements
    - jdbc:mysql:loadbalance://node1,...,nodeN/...loadBalanceStrategy =random
- Many more options

Class Not Found

- java.lang.ClassNotFoundException: com.mysql.jdbc.Driver
- The driver is not in your CLASSPATH
- Standalone applications with framework-managed CLASSPATH help with this, i.e WEB-INF/lib, or mvn or ant-managed CLASSPATH

No Suitable Driver

- SQLException: No suitable driver
  - Check your URL, compare to the documentation at http://dev.mysql.com/doc/refman/5.1/en/connector-j.html
- Your URL should look something like
  - jdbc:mysql://host:port/database[...]

Out of Memory

- java.lang.OutOfMemoryError when reading results
  - Your result set is too large, Connector/J buffers results by default
  - Most use cases don’t require large result sets
  - Those that do, process in chunks via SELECT ... LIMIT, or Statement.setFetchSize(Integer.MIN_VALUE)
- Heap dump shows many statement, result set instances
  - Not closing resources on every code path
  - Frameworks help prevent this
  - Plumb try {} catch {} finally {} throughout your code (yuck!)

PHP

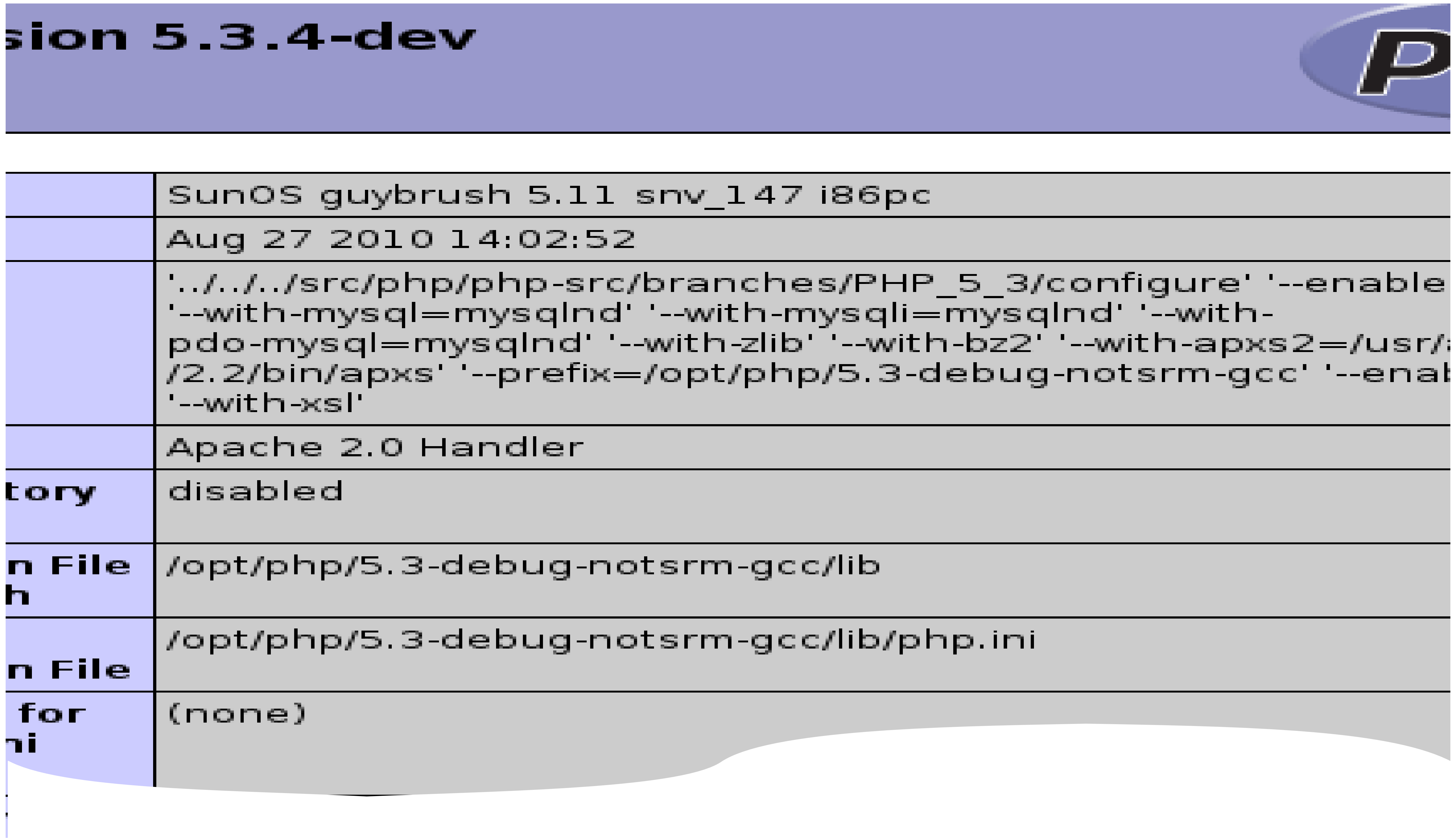
Introduction to PHP

- PHP Hypertext Preprocessor
- The most common 'P' in LAMP
- Web-Centric Scripting Language
  - Processed by a Web-Server module
  - Can be embedded in HTML
  - Built-in functionality for dealing with Web-Things
- Developed by a large Open Source community since 1995
  - Multiple Oracle employees actively involved
- PHP consists out of a relatively small core and a large collection of function libraries (“extensions”)

PHP and MySQL Resources

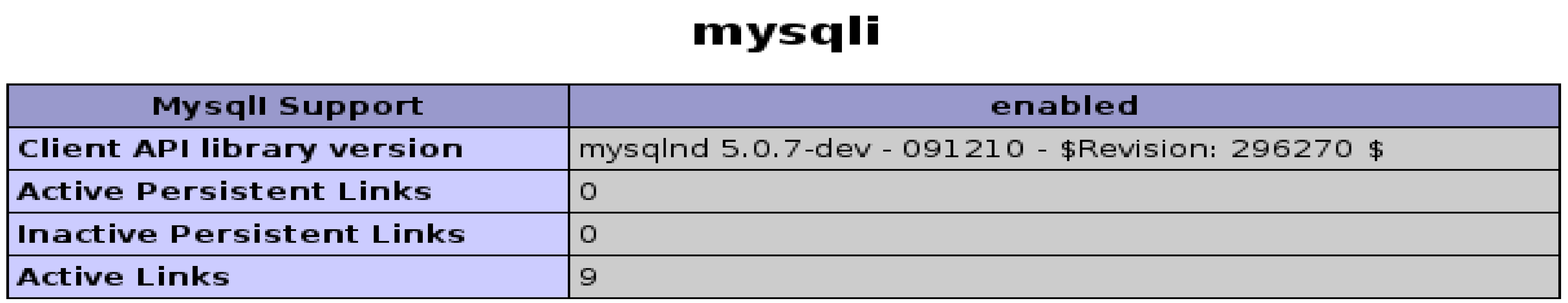
- Use MySQL with PHP
  - http://dev.mysql.com/usingmysql/php/
- Read Connector/PHP User Manual
  - http://dev.mysql.com/doc/refman/5.1/en/apis-php.html

Verifying the PHP Installation



sion 5.3.4-dev

	SunOS guybrush 5.11 snv_147 i86pc
	Aug 27 2010 14:02:52
	'../../../src/php/php-src/branches/PHP_5_3/configure' '--enable '--with-mysql=mysqlnd' '--with-mysqli=mysqlnd' '--with-pdo-mysql=mysqlnd' '--with-zlib' '--with-bz2' '--with-apxs2=/usr/ /2.2/bin/apxs' '--prefix=/opt/php/5.3-debug-notsrm-gcc' '--enab '--with-xsl'
	Apache 2.0 Handler
tory	disabled
n File h	/opt/php/5.3-debug-notsrm-gcc/lib
n File	/opt/php/5.3-debug-notsrm-gcc/lib/php.ini
for ni	(none)



mysqli

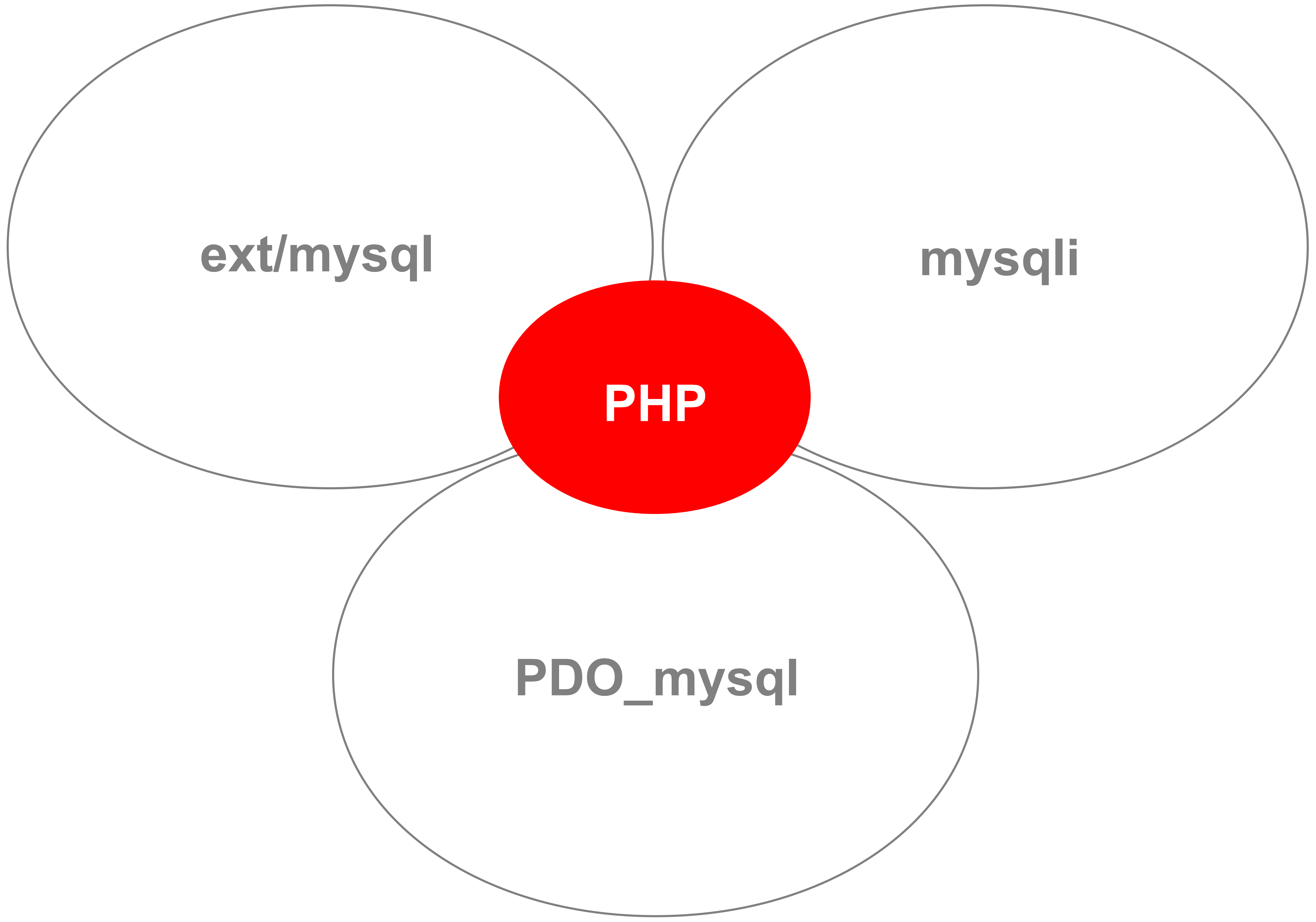
MysqlI Support	enabled
Client API library version	mysqlnd 5.0.7-dev - 091210 - $Revision: 296270 $
Active Persistent Links	0
Inactive Persistent Links	0
Active Links	9

c:\xampp\htdocs\test.php:

```
<?php
phpinfo();
?>
```

http://localhost/test.php

PHP Extensions for MySQL



ext/mysql

- One of the first PHP extensions
  - Actively maintained with PHP 4
  - No new features in PHP 5
    - Exception: Added mysqlnd support with PHP 5.3
  - Bug fixing only
- Best documented database extension
  - Tons of books, tutorials, …
- Missing support for many MySQL features
  - New Authentication protocol, Prepared statements, Queries with multiple result sets (stored procedures), compression, encryption, full charset support, …

mysqli

The Improved MySQL Extension

- Full support for all MySQL features
  - Stored Procedures
  - Prepared Statements
  - Encryption (SSL)
  - Compression
  - Charsets
  - …
- Actively developed, maintained and supported by Oracle

PDO_mysql

- “The PHP Data Objects (PDO) extension defines a lightweight, consistent interface for accessing databases in PHP.” http://php.net/intro.pdo
- Lowest common denominator
- PHPish API
- PDO is emulating prepared statements by default
  - $pdo->setOption(PDO::MYSQL_ATTR_DIRECT_QUERY, false);

ORACLE

Reasons for using different APIs

- mysqli
  - Support for all MySQL features
  - Best support / stability
  - Integration with existing applications / environments
- PDO
  - Simple applications supporting multiple databases (for instance Oracle DB and MySQL)
  - Integration with existing applications / environments

Escaping for mysqli

- mysqli_real_escape_string()
  - Escapes special characters for usage in SQL statements
  - Takes current encoding into account
  - Prevents SQL injection

```
$sql = sprintf(“INSERT INTO employees
              (birth_date, first_name, last_name, gender)
              VALUES ('%s', '%s', '%s', '%s')”,
              mysqli_real_escape_string($conn, $_POST['birth_date']),
              mysqli_real_escape_string($conn, $_POST['first_name']),
              mysqli_real_escape_string($conn, $_POST['last_name']),
              mysqli_real_escape_string($conn, $_POST['gender'])
            );
 if ( ! mysqli_query($conn, $sql) {
        // ERROR
}
```

Prepared Statements and mysqli

```
$query = "INSERT INTO employees (first_name, last_name, gender)
                VALUES (?,?,?)";
$stmt = mysqli_prepare($conn, $query);

mysqli_stmt_bind_param($stmt, "sss", $val1, $val2, $val3,$val4);

$val1 = 'Johannes';
$val2 = 'Schlüter';
$val3 = 'M';
mysqli_stmt_execute($stmt);

$val1 = 'Andrey';
$val2 = 'Hristov';
$val3 = 'M';
mysqli_stmt_execute($stmt);

mysqli_stmt_close($stmt);
```

ORACLE

PDO Basics

```
$pdo = new PDO(“mysql:host=localhost;dbname=test”,
         “user”, “password”);

 $query = $pdo->prepare(
     “SELECT id FROM table LIMIT ?, ?”);

 $query->bindValue(1, $_GET[“offset”],
     PDO::PARAM_INT);
 $query->bindValue(2, (int) $_GET[“limit”]);

 $query->execute();
```

ORACLE

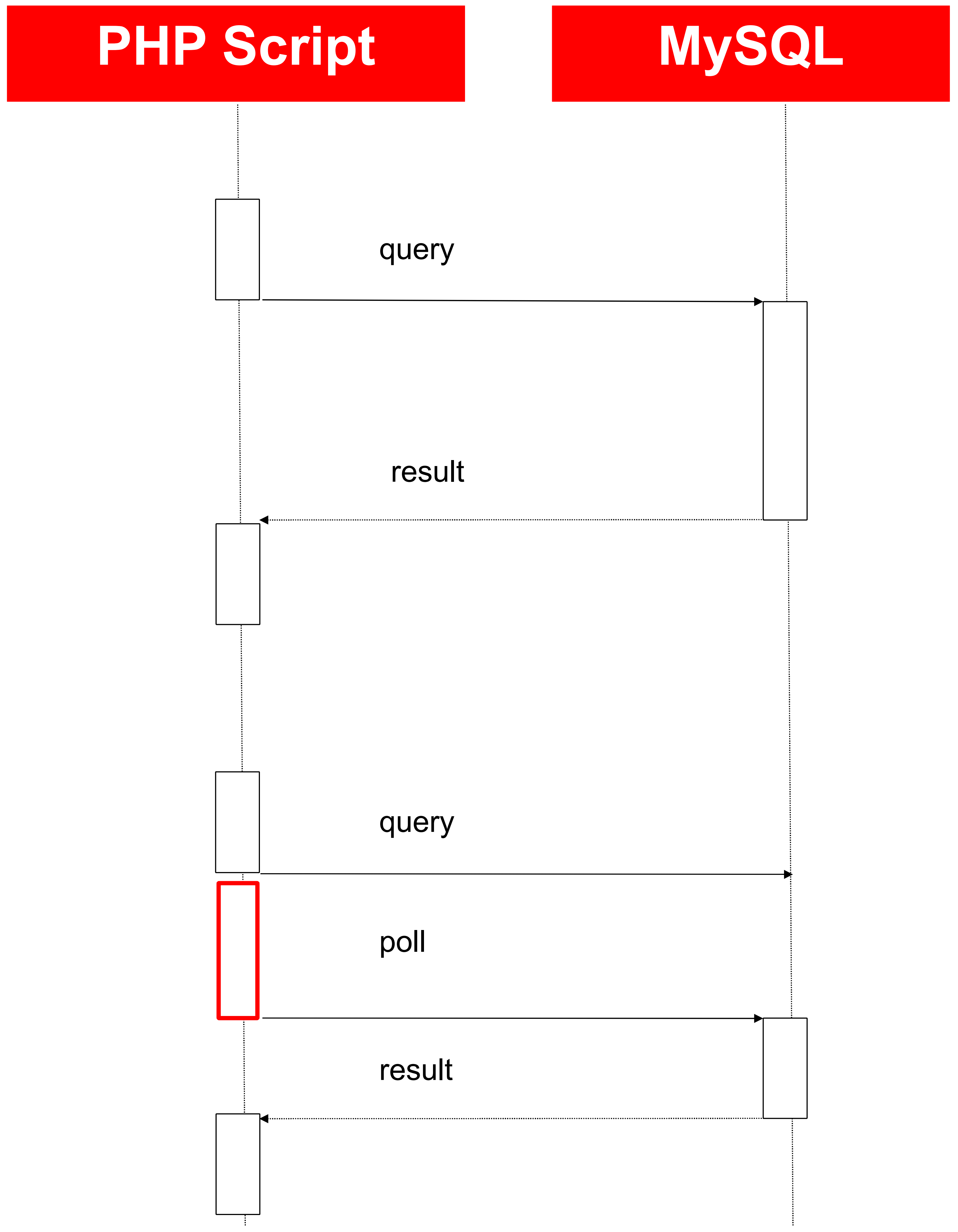
Asynchronous Queries
(mysqlnd)

```
$conn = new MySQLi(...);
$conn->query(
       "SELECT * FROM t WHERE ....",
       MYSQLI_ASYNC);


/* Do something */


mysqli_poll($links, $errors, $reject, 1);


/* Process query results */
```



.NET

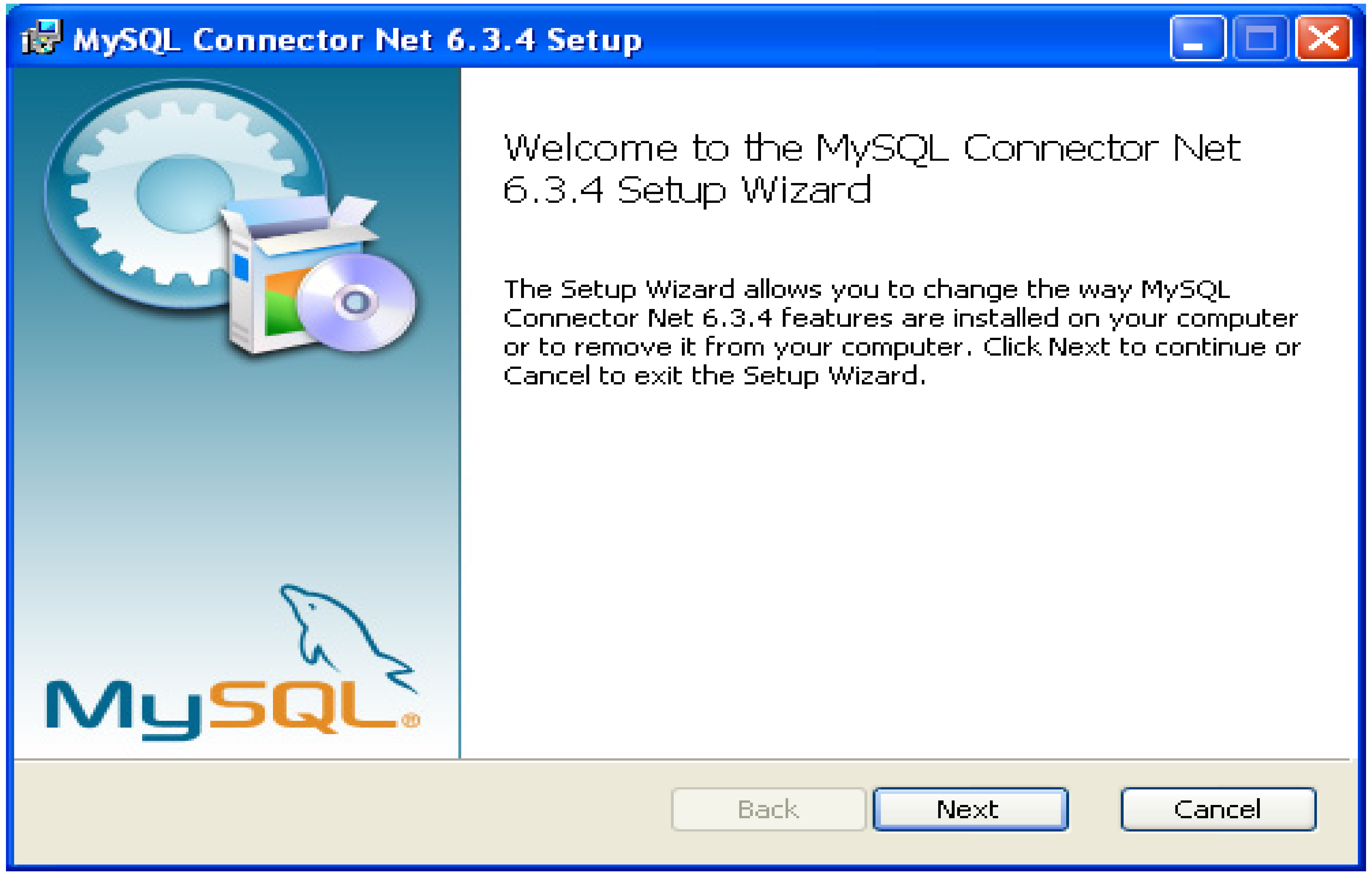
MySQL and Windows – Embrace with Windows way!

- Install Visual Studio extensions for MySQL
- Take advantage of MySQL Workbench
- Use MySQL 5.5 especially if MySQL is your server platform as well – huge performance gains!

What You Need for Connector/NET Development

- MySQL Database Installed and Running
- Connector/NET Driver (v6.3.2 or later for VS 2010)
- Visual Studio (2005, 2008, 2010)
- MySQL Workbench 5.2 (optional)

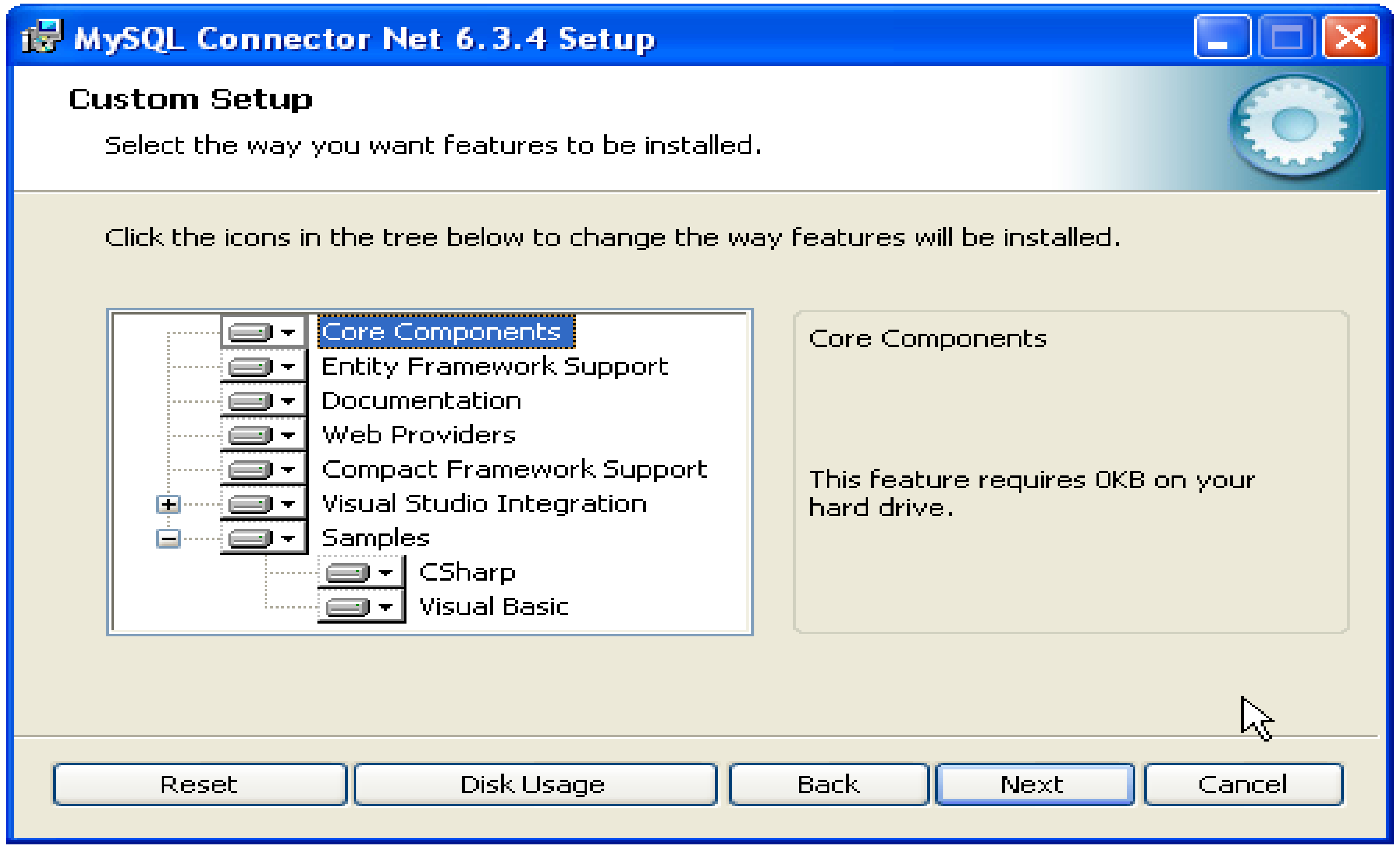
Installing Connector/NET (Windows)



MySQL Connector Net 6.3.4 Setup

Welcome to the MySQL Connector Net 6.3.4 Setup Wizard

The Setup Wizard allows you to change the way MySQL Connector Net 6.3.4 features are installed on your computer or to remove it from your computer. Click Next to continue or Cancel to exit the Setup Wizard.



MySQL Connector Net 6.3.4 Setup

Custom Setup

Select the way you want features to be installed.

Click the icons in the tree below to change the way features will be installed.

- Core Components
- Entity Framework Support
- Documentation
- Web Providers
- Compact Framework Support
- Visual Studio Integration
- Samples
  - CSharp
  - Visual Basic

Core Components

This feature requires 0KB on your hard drive.

ORACLE

Installing Connector/NET (Mono)

- There is no installer available for installing the Connector/NET component on your Unix installation. Before installing, please ensure that you have a working Mono project installation
  - You can test whether your system has Mono installed by typing:
  - shell> mono --version
  - The version of the Mono JIT compiler will be displayed.
- To compile C# source code you will also need to make sure a Mono C# compiler is installed

ORACLE

Installing Connector/NET (Mono - cont.)

- To install Connector/NET on Unix/Mono:
  - Download the mysql-connector-net-version-noinstall.zip and extract the contents to a directory of your choice, for example: ~/connector-net/.
  - In the directory where you unzipped the connector to, change into the bin directory. Ensure the file MySql.Data.dll is present.
  - You must register the Connector/NET component, MySql.Data, in the Global Assembly Cache (GAC). In the current directory enter the gacutil command:
- root-shell> gacutil /i MySql.Data.dll
  - This will register MySql.Data into the GAC. You can check this by listing the contents of /usr/lib/mono/gac, where you will find MySql.Data if the registration has been successful.

Integrating Connector/NET with Visual Studio

- MySQL Connector/NET supports Visual Studio versions 2005, 2008, and 2010. However, only MySQL Connector/NET version 6.3 fully integrates with Visual Studio 2010
- Visual Studio 2010 support was introduced with MySQL Connector/NET 6.3.2. From version 6.3.2 the connector ships with both NET 2.x and .NET 4.x versions of the Entity Framework support files, mysql.data.ef.dll and mysql.visualstudio.dll

Integrating Connector/NET with Visual Studio (cont.)

- When MySQL Connector/NET is installed on Microsoft Windows, Visual Studio integration components are also installed and initialized. This enables the developer to work seamlessly with MySQL Connector/NET in the familiar Visual Studio environment.

Integrating Connector/NET with Visual Studio (cont.)

- The .NET 4.x versions need to be shipped to enable new integration features supported in Visual Studio 2010, including:
  - New DDL T4 template for the Entity Framework (EF)
  - Enables developers to design an EF model from scratch and use the native Visual Studio 2010 facility to generate MySQL DDL from that model. This is done by creating the model and choosing the SSDLToMySQL template in the properties window.
  - The correct DDL is then generated and the developer can then save this code as a .mysql file in their project and execute it against the MySQL server.

Integrating Connector/NET with Visual Studio (cont.)

- New SQL Editor - A new SQL editor has been included that enables connections to servers to execute SQL. This is activated by creating a new file with a .mysql extension. A new template is also included to allow creation of this file type using the Visual Studio 2010 main menu item FILE, NEW.
- Note: the MySQL SQL Editor is also available in 2005 and 2008.

.NET Options for MySQL

- Similar to Connector/J, Connector/NET has many extensions to the .NET API to enhance MySQL usage Full listing is here: http://dev.mysql.com/doc/refman/5.5/en/connector-net-connection-options.html
- They include
  - Same HA/Cluster options as Connector/J
  - Using Windows Named Pipes in place of TCPIP
  - Connection pooling in the driver
  - Compression, encryption, caching options as well

Bulk loading with Connector/NET

- You can now process text file loads similar to the MySQL `LOAD DATA INFILE` statement. Assume the following file:

```
Table Career in Test Database
Name  Age Profession


Tony  47  Technical Writer
Ana 43  Nurse
Fred  21  IT Specialist
Simon 45  Hairy Biker
```

- And a corresponding 'Careers' table in MySQL

Bulk loading setup code

```
using MySql.Data;

using MySql.Data.MySqlClient;

string connStr =
"server=localhost;user=root;database=test;port=3306;password=******;";

MySqlConnection conn = new MySqlConnection(connStr);

MySqlBulkLoader bl = new MySqlBulkLoader(conn);

bl.TableName = "Career";

bl.FieldTerminator = "\t";

bl.LineTerminator = "\n";

bl.FileName = "c:/career_data.txt";

bl.NumberOfLinesToSkip = 3;
```

Uploading

```
try{

      Console.WriteLine("Connecting to MySQL...");
      conn.Open();


      // Upload data from file
      int count = bl.Load();
      Console.WriteLine(count + " lines uploaded.");
}catch (Exception ex){
                Console.WriteLine(ex.ToString());
}
```

Resources

- Upcoming MySQL live Webinar
  - http://www.mysql.com/news-and-events/web-seminars/index.html
- MySQL on-demand Webinars
  - http://www.mysql.com/news-and-events/on-demand-webinars/
- MySQL Whitepapers
  - http://www.mysql.com/why-mysql/white-papers/
- MySQL Case Studies
  - http://www.mysql.com/why-mysql/case-studies/
- Get in Touch: http://mysql.com/contact/

Hardware and Software

ORACLE®

Engineered to Work Together

We encourage you to use the newly minted corporate tagline
“Hardware and Software, Engineered to Work Together.”
at the end of all your presentations. This message should
replace any reference to our previous corporate tagline
“Software. Hardware. Complete.”

ORACLE®